



The Shortest Path

Link: <http://www.spoj.com/problems/SHPATH/en/>

Solution:

C++	http://ideone.com/uOI7ov
Java	http://ideone.com/vLk7Ex
Python	http://ideone.com/723v1S (TLE)

*** **Lưu ý** với Python: Mã nguồn tham khảo khi submit sẽ bị TLE do dữ liệu trên SPOJ chưa sẵn sàng cho ngôn ngữ này, các bạn có thể submit mã nguồn tại www.bigocoder.com.

Tóm tắt đề: Cho bạn danh sách các thành phố và các đường đi giữa các thành phố với nhau. Cho bạn danh sách mô tả các thành phố và các đường kết nối. Hãy tìm đường đi ngắn nhất từ thành phố này đến thành phố kia.

Input

Dòng đầu tiên chứa số lượng bộ test s ($s \leq 10$). Mỗi bộ test có định dạng như sau:

- Dòng đầu chứa số n ($n \leq 10.000$) – số lượng thành phố.
- n cụm tiếp theo, mỗi cụm chứa thông tin của thành phố có kết nối với các thành phố khác với thông tin như sau:
 - Tên thành phố *NAME* – ứng với số thứ tự từ 1 đến n .
 - Số nguyên p là số lượng thành phố có kết nối với thành phố *NAME*.
 - p dòng tiếp theo mỗi dòng chứa 2 giá trị: nr – giá trị thành phố có kết nối với thành phố hiện tại và $cost$ chi phí kết nối ($cost \leq 200.000$)
- Tiếp theo là một số nguyên r – số cặp các thành phố cần tìm đường đi ngắn nhất.
- r dòng tiếp theo, mỗi dòng chứa một cặp *NAME1 NAME2* là tên thành phố bắt đầu và kết thúc.
- Có dòng trống ngăn cách giữa các bộ test.**

Output

Với mỗi dòng có 2 tên thành phố cần tìm đường đi ngắn nhất bạn sẽ in ra chi phí tìm được.

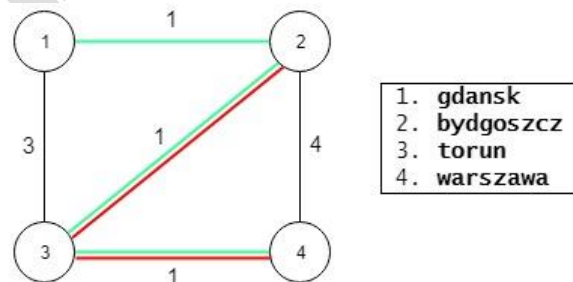
1	3
4	2
gdansk	
2	

2 1	
3 3	
bydgoszcz	
3	
1 1	
3 1	
4 4	
torun	
3	
1 3	
2 1	
4 1	
warszawa	
2	
2 4	
3 1	
2	
gdansk warszawa	
bydgoszcz Warszawa	

Giải thích: Có 4 thành phố lần lượt có tên là: gdansk, bydgoszcz, torun, warszawa.

Đường đi ngắn nhất giữa 2 thành phố:

- gdansk → warszawa (đường màu xanh) có chi phí: $1 + 1 + 1 = 3$.
- bydgoszcz → warszawa (đường màu đỏ) có chi phí: $1 + 1 = 2$.



Hướng dẫn giải: Bài này ý tưởng giải thì không khó, cái khó là bạn phải chuyển dữ liệu từ chuỗi về số để có thể chạy thuật toán Dijkstra cho từng trường hợp. Cách giải quyết như sau:

Bỏ các tên thành phố vào một mảng danh sách các thành phố (mảng chuỗi). Sau đó bỏ lần lượt từng đường đi vào graph bình thường. Khi tìm đường đi giữa 2 thành phố với nhau chỉ cần tìm lại index của thành phố đó ban đầu khi bỏ vào mảng thành phố là xong.

Độ phức tạp: $O(s * p * E \log V)$ với s là số lượng bộ test, p là số lượng truy vấn tìm đường đi ngắn nhất giữa 2 thành phố, E là số lượng cạnh (cung) của đồ thị và V là số lượng đỉnh của đồ thị trong mỗi bộ test.